A COMPREHENSIVE STUDY OF SHONAN ROTATION AVERAGING ALGORITHM FOR SOLVING ROTATION AVERAGING PROBLEM IN STRUCTURE FROM MOTION SYSTEM

A Dissertation
Presented to
The Academic Faculty

By

Jing Wu

In Partial Fulfillment
of the Requirements for the Degree
Master of Science in the
College of Computing
Department of Interactive Computing

Georgia Institute of Technology

May 2021

A COMPREHENSIVE STUDY OF SHONAN ROTATION AVERAGING ALGORITHM FOR SOLVING ROTATION AVERAGING PROBLEM IN STRUCTURE FROM MOTION SYSTEM

Thesis committee:

Dr. Frank Dellaert, Advisor
School of Interactive Computing
Georgia Institute of Technology

Dr. James Hays
School of Interactive Computing
Georgia Institute of Technology

Dr. Cedric Pradalier
School of Interactive Computing
Georgia Institute of Technology

Date approved: April 29th, 2021

博学而笃志，切问而近思

from 自由而无用的灵魂

Dedicated to my parents, Kefang Wu and Xiaomei Wen, thanks for their thorough support and love.

ACKNOWLEDGMENTS

First, I would like to thank my advisor, Dr. Frank Dellaert, who provided a great opportunity for me to study rotation averaging problem and learn more about graph optimization. His feedback and guidance throughout my graduate study were invaluable and helpful. I'm really grateful for it.

I would also like to acknowledge Dr. David Rosen for providing meaningful suggestions and helps, which set a great model for me.

Many thanks to Fan Jiang, Ayush Baid, Yetong Zhang, Varun Vagrawval, Mandy Xie, Akshay Krishnan, Abhinav Jain, Sicong Ma, Gerry Chen, and John Lambert in Borg Lab for the advice on both academic and personal life.

My work cannot be accomplished without the help from them. In the end, I owe my deepest gratitude to my family's unconditional support and my friends' backup for assisting me come across all difficulties during this special period.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

The object of the proposed work is to further understand Shonan Rotation Averaging algorithm. Rotation averaging problem is to recover the absolute camera orientations given a set of relative camera rotations. The difficulty in rotation averaging algorithm is the high dimension and non-convexity caused by large amount of camera pose number and orthogonal constraints inside rotation matrix. Shonan Averaging algorithm applies the convex relaxation to the original problem and use the duality theory to prove the approximation will generate the global optimal solution. Additionally, Shonan Averaging is able to give a global optimal certification while provides fast and accurate result. This thesis studies Shonan Averaging from several dimensions: the different initialization method, several hyper parameters inside the algorithm and comparison with other optimization methods. At the end of the thesis, I discuss some possible directions that Shonan Averaging algorithm can improve.

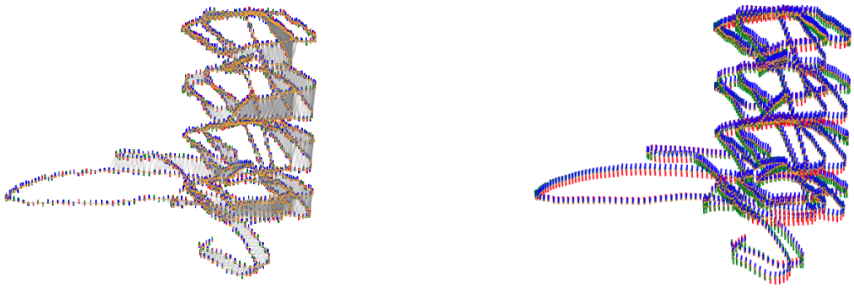
CHAPTER 1
INTRODUCTION

1.1 Rotation Averaging Problems

For artificial intelligent products like autonomous driving cars, augmented reality devices, unmanned aerial vehicles and human-like robots, perceiving the surroundings and localizing the position correctly by using Simultaneous Localization and Mapping (SLAM) and Structure from Motion (SfM) [1] are extremely significant and are usually the first step before the robot/device is able to interact with the world. Thus discovering more reliable algorithm for solving rotation averaging problem in SLAM and SfM is of high importance.

Rotation Averaging is the problem of how to recover the absolute rotation matrices of camera poses when given a set of their relative rotation measurements [2, 3]. As shown in Figure 1.1, this example of rotation averaging problem, the grey lines in the left image are the relative measurements, the orange dots are the ground truth position and the red-green-blue axes denote the absolute camera orientations. We can use rotation averaging algorithm to recover the ground truth camera rotations from randomly initialized camera rotation given the measurements between these relative camera rotations. This problem usually occurs in pose estimation process and bundle adjustment [4] process both in SLAM and SfM. SLAM and SfM are techniques for 3D surrounding structure recovering and trajectory tracking by using sensors like camera, while SLAM focuses more on simultaneous localization where photos are received in sequence and SfM focuses more on structure recovering where photos are usually received in random order. There are already several effective optimization libraries which are suitable for implementing Shonan Averaging algorithm [5, 6, 7].

In this thesis, I studied the Shonan Rotation Averaging algorithm proposed by [8] to



(a) random initialization (b) optimized estimation

Figure 1.1: Comparison between the random initialization and the optimized result by Shonan Averaging algorithm on parking garage dataset with 1661 camera poses and 6275 relative camera measurements.

have a better understanding of how this algorithm works, what is the performance of this algorithm on real-world and synthetic datasets and how to improve this algorithm in practical problem.

1.2 Difficulties in Rotation Averaging Problems

1.2.1 Maximum Likelihood Estimation

The rotation averaging problem can be present as a Maximum Likelihood Estimation problem [9, 10]:

$$\max_{R \in SO(d)^n} \sum_{i,j \in \epsilon} k_{ij} tr(R_i \bar{R}_{ij} R_j^T) \tag{1.1}$$

Maximize the trace in Equation 2.1 is equal to maximize the projection of $R_i \bar{R}_{ij}$ on R_j^T if we can visualize the latter as a coordinate with the three axes setting as the three rotation vectors in it. I followed the form in [8] to present the rotation averaging in a mathematical way. By transforming the maximum likelihood estimation of the rotation

averaging problem into minimizing the sum presented in:

$$\min_{R \in SO(d)^n} \sum_{i,j \in \epsilon} k_{ij} \|R_i - R_i \bar{R}_{ij}\|_F^2 \tag{1.2}$$

where $k_{ij} \geq 0$ are parameters for Langevin noise model [11, 12], $R_1, \cdots, R_n \in SO(d)$ are the orientations and $\bar{R}_{ij} \in SO(d)$ are the relative measurements, which is supposed to be $R_i^{-1} R_j$.

1.2.2 Extrinsic and Intrinsic error

There are two forms of representation for the error in the cost function, the extrinsic metric that can show the error not in the manifold space and the intrinsic metric that can show the geometrical distance on the manifold, for example, the angle error between two rotations.

Here we use the Frobenius matrix norm, also known as extrinsic metrics [13] as the cost during optimization. The Frobenius matrix norm represents the distance through non-manifold space. Since there is dimension lifting in Shonan Averaging algorithm, it is more consistent to calculate the error between the estimation and measurements using extrinsic metrics.

Another way is to calculate the Euler angle error of two rotation matrices as in Equation 1.3, which is the geodesic distance in the rotation manifold. This is also equal to maximize the trace in Equation 1.1 (maximize the projection).

$$\min_{R \in SO(d)^n} \sum_{i,j \in \epsilon} k_{ij} \angle (R_i^{-1} R_j)^T \bar{R}_{ij} \tag{1.3}$$

1.2.3 Hardness in optimization

The rotation averaging problem is hard to solve due to the inner orthogonal constraints of the rotation matrix. The objective function in Equation 1.2 has non-convexity due to the non-convexity on the manifold of rotation. When the camera number n increases, this

problem is at high dimension, which makes it harder to solve [8].

1.3 Existing Solutions for Rotation Averaging

To relax the non-convexity of rotation averaging problem, there are several typical methods.

1.3.1 Relaxed method

Instead of using rotations on special orthogonal group, the relaxed approach using quaternion [14] to make the objective function linear. But the solution in quaternion form might not satisfy the unit constraint. The solution is needed to be rounded in order to satisfy this constraint. The rounded solution can have relaxation gaps due to the rounding and thus it is not a guaranteed optimal solution.

1.3.2 Duality constraint

Another method using Lagrangian duality principle is proposed by Eriksson [15]. By approximating the original problem with a dual problem which is a convex semidefinite program, this algorithm then uses constant block diagonals for solving the semidefinite program. If the noise is under considerable level, then the strong duality will always hold and global optimal is guaranteed.

1.3.3 Shonan

Shonan Rotation Averaging algorithm [8] combines the advantage of the two methods. It uses dimension lifting and duality theory to relax the non-convexity on the original rotation manifold. By approximating the original problem with proven relaxed program, Shonan Averaging can get a good first-order initial estimation on the rotation manifold. Then it passes the first-order solution to a traditional optimizer like Levenberg Marquardt (LM) [16] and searches for the second-order solution. With the good initial estimation provided by solution of the dual problem, using traditional methods like LM can provide the local

minimum around this initial estimation. Shonan also provide a global optimal certification for verification.

1.4 Datasets

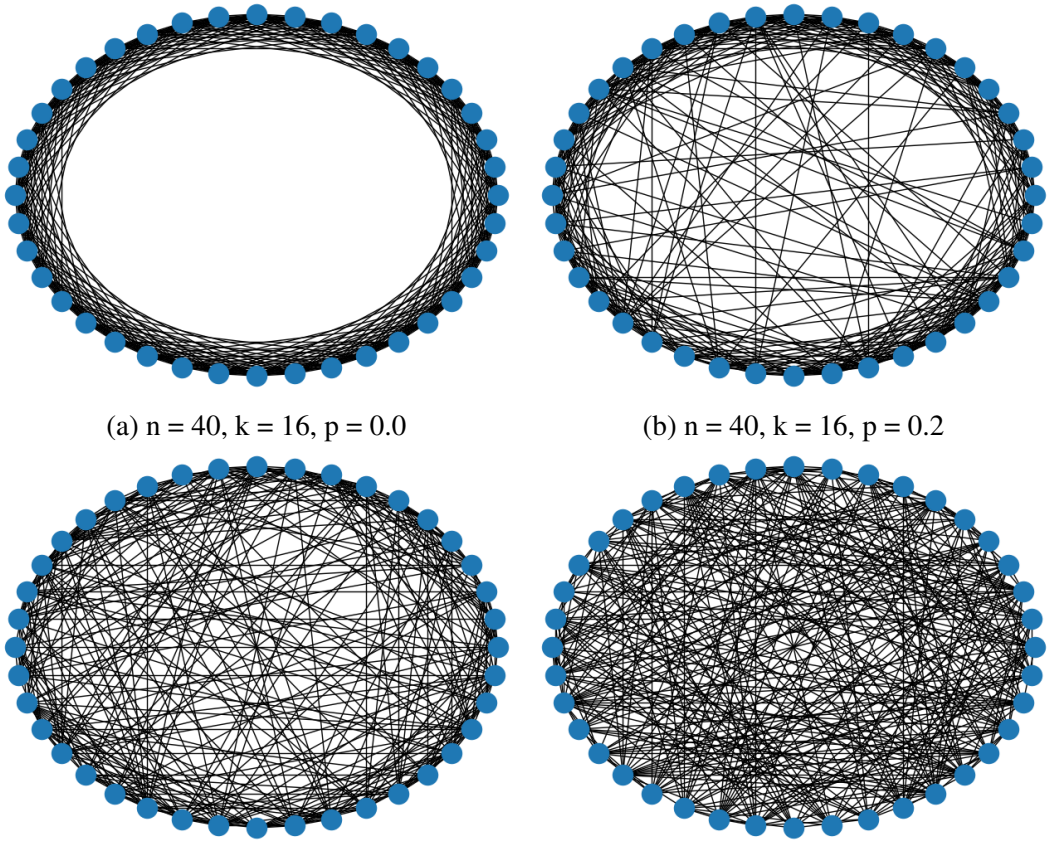
To test the performance of Shonan Rotation Averaging, we utilized synthetic data and real-world data to study the performance and computation efficiency of Shonan Averaging and other exsisting nonlinear optimization methods mentioned above.

1.4.1 Watts-Strogatz Graph Model

The synthetic datasets are generated by utilizing the Watts-Strogatz graph model [17]. The Watts-Strogatz model is often used in random graph problem generation. When given the number of nodes n, the mean degree k of each node and a probability parameter p where $0 \leq p \leq 1$ and $n \gg k \gg \ln n \gg 1$, the model generate a undirected graph with n nodes and $\frac{nk}{2}$ edges.

There are two steps in total to construct a Watta-Strogatz graph. 1) Construct a graph with the nodes in a circular way, each node is connected the k neighbored nodes, $\frac{k}{2}$ on each side. 2) For each node, rewire the right side edges with probability p by replacing the original edge (i, j) where $i < j \leq i + \frac{k}{2}$ with a uniformly random sampled edge (i, k) where $k \neq i$ and (i, k) does not duplicate with already existed edges.

We implemented this model by using the connected_watts_strogatz_graph method in **Networkx** (python) to ensure the graph is connected. As shown in Figure 1.2, there are examples of four small-world graph generated by the Watts-Strogatz model. While keeping the node and edge number 40 and 320, we increase the rewiring probability from 0.0 to 1.0. As the rewiring probability increases, which means the edges connected with its k neighbors are more likely to rewire with other nodes, the graph becomes more connected and we will show that the connectivity of the graph will have a impact on the difficulty of solving the rotation averaging problem in the experimental part later.



(a) n = 40, k = 16, p = 0.0

(b) n = 40, k = 16, p = 0.2

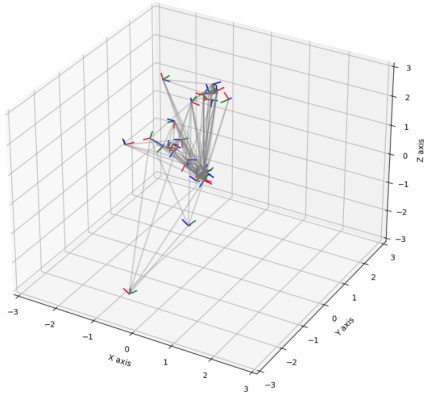
(c) n = 40, k = 16, p = 0.5

(d) n = 40, k = 16, p = 1.0

Figure 1.2: Visualization of Watts-Strogatz small world datasets with different rewiring probabilities.

1.4.2 YFCC datasets

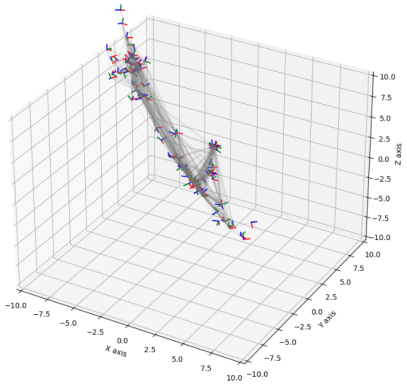
For the real-world datasets, we choose the Yahoo Flicker Creative Commons 100 Million Datasets (YFCC100M) [18]. We used the Deep Front-Ends [19] to extract the relative geometry between camera poses of the images selected in the YFCC100M datasets. We selected datasets with camera pose number ranging from 10 to 2000, relative camera measurement ranging from 50 to 100000. Most of the datasets are top scenic spots, e.g. the Natural History Museum London, Notre Dame Rosary Window, etc. Figure 1.3 is the visualization for some of the YFCC datasets. The orientations are the absolute camera poses in these datasets that are estimated by the [19] and the relative measurements (not shown in Figure 1.3) are also provided by the [19].



(a) Natural History Museum of London

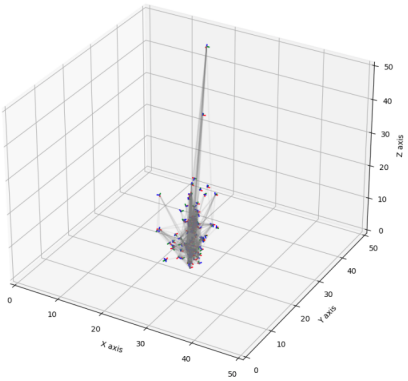
(b) Example image of Natural History Museum of London

Figure 1.3: Visualization of absolute camera poses for four YFCC datasets.



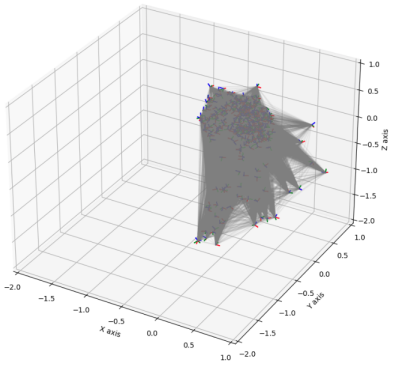
(c) London Bridge

(d) Example image of London Bridge



(e) Lincoln Memorial

(f) Example image of Lincoln Memorial



(g) Notre Dame Rosary Window

(h) Example image of Notre Dame Rosary Window

Figure 1.3: Visualization of absolute camera poses for four YFCC datasets.

1.5 Overview

In chapter 2, I describe the optimization procedure of Shonan Averaging Algorithm in a mathematical way, following the approach in [8]. Since we use GTSAM [7] as the base optimization library, I will introduce the graph optimization theory and the connection between graph and rotation averaging. In our experiments, I notice that the performance of rotation averaging algorithm differs when given different initialization and I will elaborate more on this point.

In the following Chapters, I study the performance of Shonan Averaging algorithm along two dimensions: the type of initial estimation given to the algorithm and the hyper parameters inside the algorithm. There is also statistics that visualize the timing result for each part inside Shonan Averaging algorithm, which provided the insights about what can be improved in the future to make Shonan faster.

In chapter 3, I demonstrate the performance of Shonan Averaging Algorithm by studying how the initialization of estimation will influence the performance of Shonan Averaging.

In chapter 4, I study how the hyper parameters will influence the accuracy and speed of Shonan Averaging algorithm.

In chapter 5, I also compare the performance of Shonan Averaging algorithm with other rotation averaging algorithms mentioned above. I compare Shonan Averaging algorithm with other optimization algorithm mentioned above using different datasets.

The content of how to make this algorithm performs best in practice is presented in chapter 6.

CHAPTER 2

SHONAN AVERAGING

Shonan Averaging Algorithm is able to recover a global optimal solution even with a bad initial estimate while providing a certificate to check global optimal at the same time. In this Chapter, I will explain the detail of Shonan Averaging algorithm. In sum, Shonan Averaging algorithm use approximation to achieve Semi-Definite Programming (SDP) for relaxing the non-convex problem and use duality theory to generate the certificate for guarantee the global optimal of the solution.

After we get the first order solution by the relaxed form of original problem, we use GTSAM [7] as the base library as the optimizer to further optimize the solution. GTSAM is a C++ library using graph optimization theory.

The complete Shonan Averaging algorithm was described in our paper published at [8] which includes more detailed proof and referenced theorems. I do not reproduce the entire algorithm here in its full detail. However, below I review some of the key calculations that affect performance.

2.1 Convex relaxation

A more compact form of the rotation averaging problem is:

$$f^*_{MLE} = \min_{R \in SO(d)^n} tr(\bar{L} R^T R^\intercal) \tag{2.1}$$

where $\bar{L}$ is a $(d \times d)$-block-structured matrix called the connection Laplacian [20] constructed by the measurements $\bar{R_{ij}}$, d usually equals to 2 or 3 depending on which dimension the rotations are from.

According to the definition of rotation matrix [21], $R_i^T R_i = \mathbf{I}^d, R_i \in SO(d)$. We can

replace the product with a Semi-Definite matrix Z, which is a $(d \times d)$-block-structured matrix constructed by $\mathbf{I_{nd \times nd}}$ to give convex relaxation to this optimization problem. Therefore this problem is also guaranteed to have a global optimal solution. After proving $Z = R^{*\intercal} R^{*T}$ and $Z = S^{*\intercal} S^{*T} (S \in \mathbb{R}^{p \times nd}, p \ll nd)$ have equally same solution, we can rewrite the problem as:

$$f^*_{SDPLR} = \min_{S \in St(d,p)^n} tr(\bar{L} S^\intercal S^T) \tag{2.2}$$

2.2 Second order direction

Let S^* be the first-order critical point of Equation 2.2, how can we find the second-order critical point. We can construct a matrix called certificate matrix [22, 23], which is:

$$C \triangleq \bar{L} - \frac{1}{2} \left(\bar{L} S^\intercal S^T + S^\intercal S^T \bar{L} \right), S \in St(d,p)^n, St(d,p) \in \mathbb{R}^{p \times d} \tag{2.3}$$

By computing the minimum eigenvalue of C, we can identify whether S^* is a global optimal solution. If $\lambda_{min} \geq 0$, then S^* is a global minimizer of Equation 2.2. If $\lambda_{min} < 0$, we can lift up the dimension of S^* by simply adding a row of zero at the end and find the second-order descent direction $\dot{S}^+$.

Here we use the Stiefel manifold [24] to generate the certificate to check global optimal and find the descent direction in the next higher level, but in practice, we still use $SO(p)$ form in optimizer. It is easy to change from $SO(p)$ to $St(d,p)$, we only need a projection matrix $P = \{\mathbf{I}_d, \mathbf{0}\} \in \mathbb{R}^{d \times p}$. The method of adding the second-order descent direction information to the higher level $p+1$ is to round the solution of $S^* = \{S_1, S_2, \ldots, S_n, S_i \in St(d,p)\}$ to $R^* = \{R_1, R_2, \ldots, R_n, R_i \in SO(p+1)\}$ using the second order direction [25] information given by the minimum eigen vector.

Suppose S_i is constructed by $R_i(p)$ by adding a row of zero at the end, we use S^* constructed by S_i to compute equation (Equation 2.3), if $\lambda_{min} \geq 0$, it tells us the S^* is the

global solution of problem Equation 2.2, we can project S_i onto $R_i(p)$ by multiplying S_i with the projection matrix P.

If $\lambda_{min} < 0$, we can use the eigen vector $v = \{v_1, v_2, \ldots, v_n, v_i \in \mathbb{R}^{1\times d}\}$ belongs to the λ_{min}, and get the descent direction $\dot{R}_i(p+1)$ in tangent space of $\dot{R}_i(p+1)$, where $\dot{R}_i(p+1)$ is:

$$\dot{R}_i(p+1) = \begin{bmatrix} 0 & -v_i \\ v_i{}^T & 0 \end{bmatrix} \in T_{R_i(p+1)}(SO(p+1)) \tag{2.4}$$

$$R_i(p+1) = \begin{bmatrix} R_i(p) & 0 \\ 0 & 1 \end{bmatrix} \tag{2.5}$$

Then we can get the initial estimation $\dot{R}_i{}^{'}(p+1)$ for the next iteration, where $\dot{R}_i{}^{'}(p+1)$ is:

$$\dot{R}_i{}^{'}(p+1) = R_i(p+1) \cdot \dot{R}_i(p+1) \tag{2.6}$$

We need to repeat this process until $\lambda_{min} \geq 0$, and then we can make sure we get the global optimal solution of both the (Equation 2.2) and (Equation 2.1).

2.3 Exponential Mapping

To do the multiplication in the Equation 2.6, we need to fill the tangent vector with the information provided by the eigen vector of minimum eigen value, then use the skew matrix to transform the direction of the tangent vector into a $SO(p+1)$ form and use the Caylay map [26] to realize Equation 2.6. For rotation in $SO(p)$ form, the tangent vector $\omega = \{\omega_{\frac{p(p-1)}{2}}, \omega_2, \ldots, \omega_1\} \in \mathbb{R}^{\frac{p(p-1)}{2}\times 1}$; for rotation in $SO(p+1)$ form, the tangent vector $\omega^{'} = \{\omega_{\frac{p(p+1)}{2}}, \omega_2, \ldots, \omega_1\} \in \mathbb{R}^{\frac{p(p+1)}{2}\times 1}$.

The skew matrix [27] of ω is:

$$\xi_{skew} = \begin{bmatrix} 0 & -\omega_{\frac{p(p-1)}{2}} & \omega_{\frac{p(p-1)}{2}-1} & \dots \\ \omega_{\frac{p(p-1)}{2}} & 0 & \dots & \\ -\omega_{\frac{p(p-1)}{2}-1} & \dots & 0 & \\ \dots & & & 0 \end{bmatrix} \in \mathbb{R}^{p\times p}$$

While the skew matrix of $\omega^{'}$ is:

$$\xi^{'}_{skew} = \begin{bmatrix} 0 & -\omega_{\frac{p(p+1)}{2}} & \omega_{\frac{p(p+1)}{2}-1} & \dots \\ \omega_{\frac{p(p+1)}{2}} & 0 & \dots & \\ -\omega_{\frac{p(p+1)}{2}-1} & \dots & 0 & \\ \dots & & & 0 \end{bmatrix} \in \mathbb{R}^{(p+1)\times(p+1)}$$

We need to fill the first 3 elements of $\omega^{'}$ with the eigen vector for the information of the increased dimension is always in the top of the tangent vector. Then we can get the exact mapping from $\frac{p(p+1)}{2}$-dimensional increment $\omega^{'}$ onto the proper rotation manifold $SO(p+1)$.

$$\exp^{\xi^{'}} \approx \left(\mathbf{I} + \xi^{'}_{skew}\right)\left(\mathbf{I} - \xi^{'}_{skew}\right) \tag{2.7}$$

CHAPTER 3
THE INFLUENCE OF INITIALIZATION METHODS AND DATASETS

In this Chapter, I study the influence of initialization methods and datasets to Shonan Averaging algorithm. For different datasets, the way the measurements are connected with the node will have an impact on the difficulty of the rotation averaging problem that the dataset forms. In section 3.1, graph connectivity is used as the standard to describe this property. Except for the the dataset itself, the initialization given to the algorithm will influence the performance of the algorithm. I discuss the influence of different initialization methods in the following sections.

3.1 Graph Connectivity

As mentioned above, we use GTSAM as the base library of Shonan Averaging algorithm. GTSAM is a C++ library using factor graph models proposed by [28]. A factor graph is a bipartite graph composed with nodes and edges, where the nodes are the unknown variables we want to estimate and the edges are the factors representing the connection and constraints between the nodes derived from our measurement. The rotation averaging problem can be transformed into a factor graph problem. In the graph of a rotation averaging problem, the factors are the relative rotations between all camera poses and the variables are the absolute camera rotations we want to get solved.

In [13] and [29], the problem of what kind of rotation averaging problem should be recognized as hard is proposed, leading research to the structure of the graph, the density, connectivity and clustered, etc. I have elaborate some on the Watts-Strogatz graph model in subsection 1.4.1. In this model, we can control the connectivity of the graph by changing the rewiring probability of the graph. The lower the rewiring probability p is, the less connected the graph is; the higher the rewiring probability p is, the more connected the

graph is.

As shown in Table 3.1 (all timing results are generated by the machine in Appendix A), I generated several datasets of varying size and rewiring probability using the Watts-Strogatz graph model and give Shonan Averaging randomly generated initialization. The camera poses are generated in a circular way, with the angles between two consecutive nodes equal and the edges are generated by the Watts-Strogatz graph model. There are three sizes of datasets, with camera poses ranging from 40 to 500. As the rewiring probability increases, the graph connectivity [30] also increases. Shonan Averaging algorithm will generate the solution with cost very close to zero. There is a time gap between the probability 0.0 and 0.2, as the graph connectivity increases, the time taken for solving the problem decreases because the problem now become more convex for the solver.

In Figure 3.1, the relation between the graph connectivity can be shown more clear. The color represents different camera sizes, the y-axis is the time taken for Shonan Averaging algorithm to solve the problem and the x-axis is the graph connectivity of the dataset. For datasets with the same camera pose numbers, increase the rewiring probability will make the graph more connected and make the problem easier to solve. This reduction in time becomes more obvious as the camera pose number increases. When the graph is already densely connected, increase the rewiring probability will mot have much improvement and the reduction in time becomes minor.

3.2 Initialization Methods for Rotation Averaging

In the following parts, I study the influence of three initialization methods: random, chordal and minimum spanning tree initialization. I experiment Shonan Averaging algorithm with different initial estimations and test both the accuracy and speed performance.

Table 3.1: Performance of Shonan Averaging algorith on Watts-Strogatz graph model datasets with different graph connectivity.

n,m	probability	init	graph connectivity	cost	total time (s)
n=40 m=320	p=0.0	random	4.607	0.000	0.107
n=40 m=320	p=0.2	random	6.401	0.000	0.040
n=40 m=320	p=0.5	random	9.589	0.000	0.048
n=40 m=320	p=1.0	random	9.951	0.000	0.045
n=100 m=2000	p=0.0	random	10.428	0.000	0.756
n=100 m=2000	p=0.2	random	18.407	0.000	0.271
n=100 m=2000	p=0.5	random	26.590	0.000	0.280
n=100 m=2000	p=1.0	random	28.564	0.000	0.274
n=500 m=50000	p=0.0	random	49.328	0.000	30.295
n=500 m=50000	p=0.2	random	94.601	0.000	11.955
n=500 m=50000	p=0.5	random	149.677	0.000	11.232
n=500 m=50000	p=1.0	random	166.001	0.000	11.576

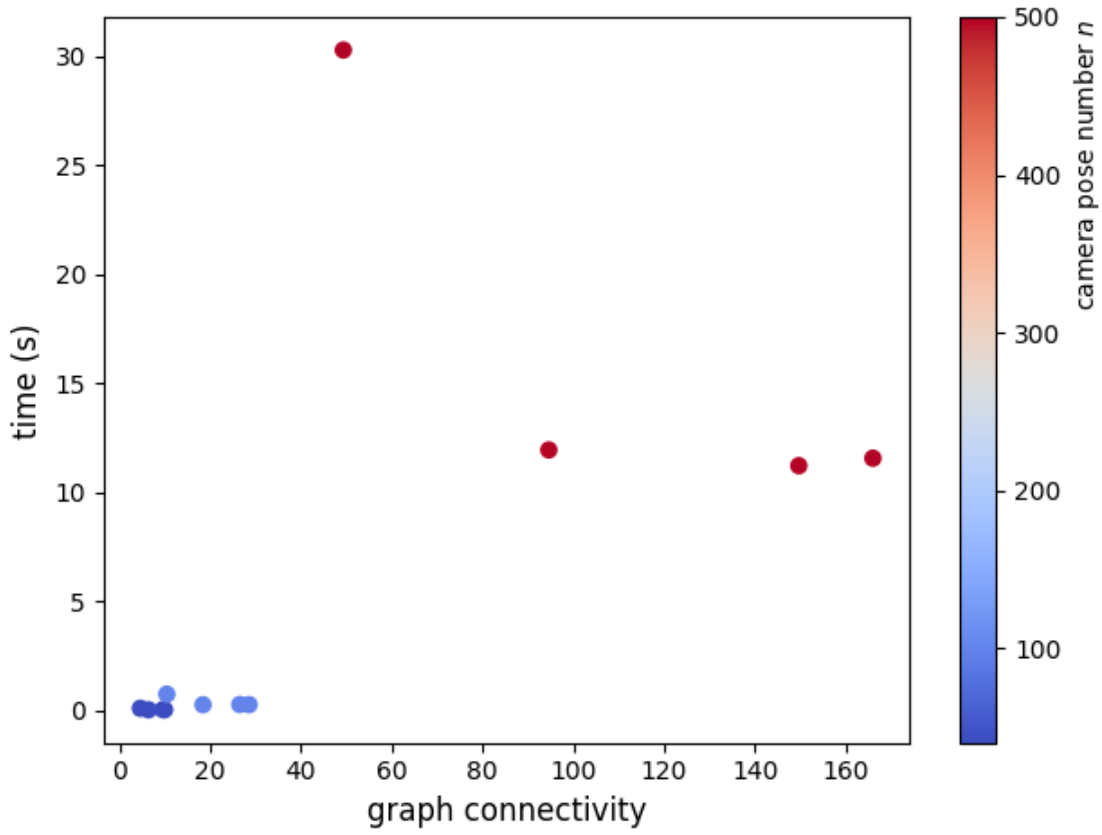


Figure 3.1: Relation between the graph connectivity and computation time of Shonan Averaging algorithm with random initialization.

3.2.1 Random Initialization

Random initialization is generated by setting all camera poses initialized with totally random orientations. This initial estimation does not contain any prior knowledge about what the absolute camera orientations might be. Thus this initialization has the largest initial error. Due to the randomness, it can distinguish the performance of different methods to the greatest extent.

3.2.2 Chordal Initialization

Chordal initialization [31] is another initialization method extracting the sub-graph of the original graph by modeling the prior factors in the graph. An anchor node is the node in the graph with highest degree and a prior factor connected with this node will be added to the graph. The sub-graph is extracted from the graph with prior factor added to the anchor node. The measurements will be regularized and added to a relaxed graph. The chordal initialization comes from the normalized results of the solution of the relaxed factor graph.

3.2.3 MST Initialization

Minimum spanning tree (MST) initialization is generated by using algorithm for finding the minimum spanning tree of the graph. After recover the initial estimation from the minimum spanning tree, noise is added to the recovered estimation for emulating the possible estimation the SfM pipeline can provide. In the experiment, we tested the datasets generated by Watts-Strogatz graph model and YFCC datasets. The noises in the measurements are equally distributed thus the graph is not weighted. In real worlds application, the noise level in the measurements should vary from each other and the graph will be weighted.

Table 3.2: Performance of different initialization on synthetic datasets generated by Watts-Strogatz graph model.

n,m	init	graph connectivity	cost	init (s)	run (s)	total time (s)
n=40 m=320	random	4.607	0.000	0.031	0.127	0.157
	chordal	4.607	0.000	0.004	0.013	0.017
	mst	4.607	0.000	0.006	0.013	0.019
n=40 m=320	random	6.401	0.000	0.002	0.050	0.052
	chordal	6.401	0.000	0.004	0.016	0.020
	mst	6.401	0.000	0.005	0.015	0.021
n=40 m=320	random	9.589	0.000	0.001	0.046	0.047
	chordal	9.589	0.000	0.004	0.013	0.018
	mst	9.589	0.000	0.005	0.012	0.018
n=40 m=320	random	9.951	0.000	0.002	0.051	0.053
	chordal	9.951	0.000	0.004	0.013	0.017
	mst	9.951	0.000	0.005	0.013	0.018
n=100 m=2000	random	10.428	0.000	0.003	0.295	0.298
	chordal	10.428	0.000	0.052	0.124	0.175
	mst	10.428	0.000	0.062	0.165	0.227
n=100 m=2000	random	18.407	0.000	0.005	0.240	0.245
	chordal	18.407	0.000	0.093	0.111	0.203
	mst	18.407	0.000	0.061	0.107	0.168
n=100 m=2000	random	26.590	0.000	0.005	0.251	0.256
	chordal	26.590	0.000	0.085	0.104	0.189
	mst	26.590	0.000	0.062	0.104	0.166
n=100 m=2000	random	28.564	0.000	0.005	0.371	0.376
	chordal	28.564	0.000	0.084	0.115	0.199
	mst	28.564	0.000	0.071	0.127	0.198
n=500 m=50000	random	49.328	0.000	0.012	27.704	27.716
	chordal	49.328	0.000	2.069	3.891	5.960
	mst	49.328	0.000	1.019	4.023	5.042
n=500 m=50000	random	94.601	0.000	0.014	11.405	11.419
	chordal	94.601	0.000	2.642	3.926	6.568
	mst	94.601	0.000	0.862	3.736	4.599
n=500 m=50000	random	149.677	0.000	0.013	10.122	10.135
	chordal	149.677	0.000	2.726	3.649	6.375
	mst	149.677	0.000	0.851	3.707	4.558
n=500 m=50000	random	166.001	0.000	0.012	12.564	12.577
	chordal	166.001	0.000	2.271	3.454	5.725
	mst	166.001	0.000	0.870	3.656	4.526

Table 3.3: Performance of different initializations on real world datasets provided by YFCC datasets.

n,m	init	graph connectivity	cost	init (s)	run (s)	total time (s)
n=30 m=274	random	3.942	0.000	0.002	0.041	0.043
natural_history	chordal	3.942	0.000	0.004	0.014	0.018
museum_london	mst	3.942	0.000	0.005	0.013	0.018
n=88 m=1500	random	1.973	0.000	0.003	0.230	0.233
london_bridge	chordal	1.973	0.000	0.019	0.130	0.149
	mst	1.973	0.000	0.048	0.097	0.145
n=127 m=3516	random	1.990	0.000	0.003	0.663	0.666
lincoln_memorial	chordal	1.990	0.000	0.048	0.183	0.231
	mst	1.990	0.000	0.116	0.210	0.326
n=173 m=11688	random	3.990	0.000	0.004	1.861	1.865
st_peters	chordal	3.990	0.000	0.183	0.721	0.904
basilica_interior	mst	3.990	0.000	0.222	0.799	1.021
n=265 m=53242	random	1.998	0.000	0.009	8.580	8.589
pike_place_market	chordal	1.998	0.000	1.128	3.569	4.697
	mst	1.998	0.000	0.941	3.518	4.460
n=326 m=93104	random	3.997	0.000	0.009	16.235	16.244
notre_dame	chordal	3.997	0.000	1.991	6.079	8.070
rosary_window	mst	3.997	0.000	1.440	4.901	6.342

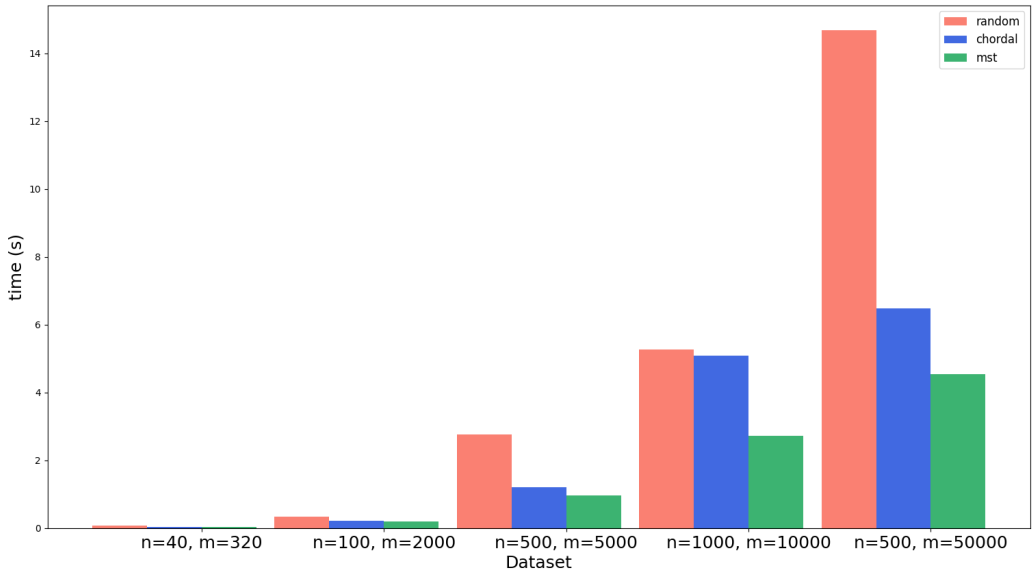


Figure 3.2: Speed performance using different initialization, random, chordal and MST on Shonan Averaging algorithm on Watts-Strogatz datasets.

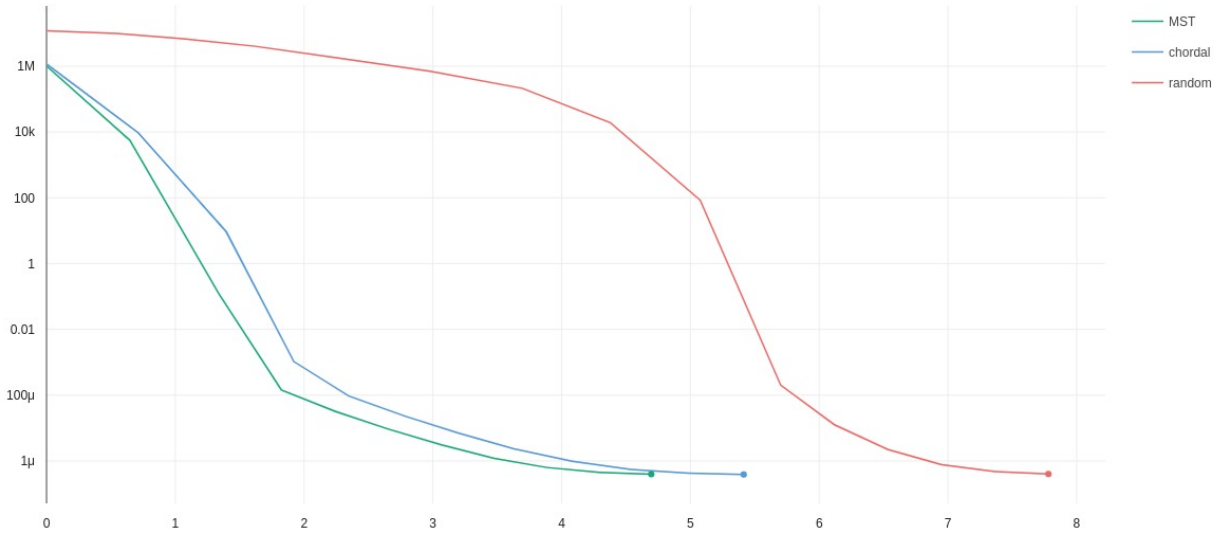


Figure 3.3: Cost change by time using different initialization, random, chordal and MST on Shonan Averaging algorithm on YFCC datasets, British Museum with 344 variables and 45450 measurements.

3.3 Influence of initialization in Watts-Strogatz graph modeled datasets and YFCC datasets

As shown in Table 3.2 and Table 3.3, using random or chordal or MST initialization will generate solutions with same cost, which further proves the global optimal of solutions returned by Shonan Averaging algorithm. n is the number of camera poses, m is the number of relative camera pose measurements, p is the rewiring probability for the Watts-Strogatz graph problem, cost is the Frobenius error of the solution, init(s) is time taken for generating the initialization, run(s) is time taken for solving the problem with given initialization, total(s) is the sum of initialization time and run time.

Using different initialization will generate same solution because they are all near the global minimum point. The difference in the three initialization methods is the speed. In general, using MST initialization can speed up the optimization process a lot by 2 to 6 times observed in the experiments.

Generating the MST initialization will take longer time than the random initialization. But the time for generating the initial estimation is minor compared with the optimization process. The MST initial estimation can provide a much better starting point for the inner LM optimizer. Thus the optimization process can speed up.

3.3.1 Speed difference between random, chordal and MST initialization

The difference of speed performance for different initialization is more clear as illustrated by Figure 3.2. I select 5 typical Watts-Strogatz graph modeled datasets with the rewiring probability setting to $p = 0.2$. As the size of the datasets grows, the difference in speed performance enlarges between random and chordal or MST initialization. In general, random initialization takes the longest time to compute. Chordal and MST initialization has faster speed while MST performs slight better than chordal initialization.

3.3.2 Accuracy difference between random, chordal and MST initialization

Figure 3.3 is a detailed visualization of the cost change during optimization for random, chordal and MST initialization, the y-axis is the Frobenius error and x-axis is the time in seconds. The y-axis is in log scale. Therefore the ultimate error is extremely close to each other. The ultimate error for random initialization is $400n$, chordal initialization is $385n$ and MST initialization is $395n$. Even with multiple times of testing, the ultimate error for three initialization always fluctuating round $390n$. This further proves the global minimum is guaranteed by Shonan.

In a real application, e.g. structure from motion problem, this pre-filtering method for providing a MST or chordal initial estimation is practical because even take the extra time for generating the more complicated initial estimation the total time for solving the rotation averaging problem is much less. By using algorithm for finding minimum spanning tree of graph, the optimization process for rotation or translation can be accelerated.

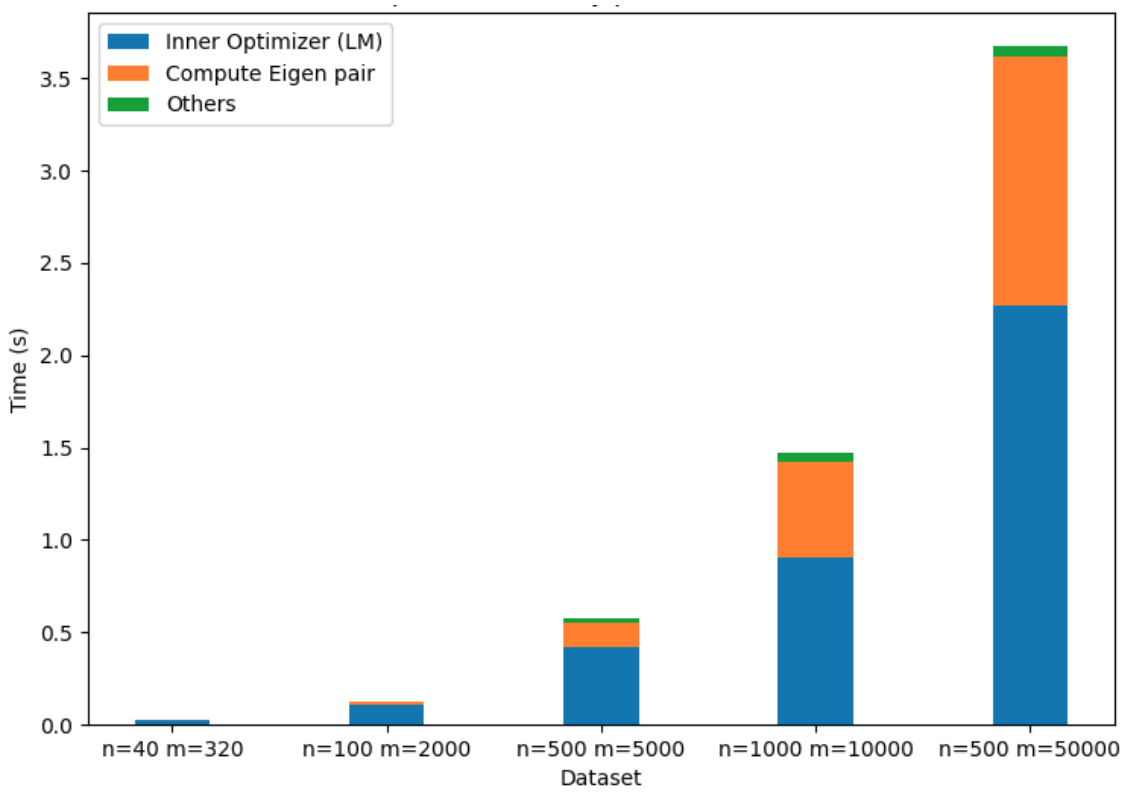


Figure 3.4: Detailed timing results of SA on Watts-Strogatz datasets

3.4 Detailed Timing results

As described in chapter 2, there are two loops inside Shonan Averaging algorithm. The first loop is for finding the first order solution by solving the Equation 2.2 on Stiefel manifold. Due to the Semi-Definite relaxation, there is no orthogonal constraints, making it possible for finding the first order solution in a fast and elegant way. The first order solution also provides insights into which direction to find the second order solution by computing the eigen vector of Equation 2.3 as the direction. After mapping the eigen vector on the rotation manifold and perturb the first order solution with this second order direction, this processed initial estimation is passed to traditional nonlinear optimizer (we use LM optimizer in our implementation), which will be the second loop in our algorithm.

In the Figure 3.4, I analysis the speed performance of Shonan Averaging algorithm with details. The test datasets are generated by the Watts-Strogatz model and initial estimation

is generated by the minimum spanning tree in the graph. The compute eigen pair time is used by the first loop for generating the eigen value for global optimal certification and eigen vector for direction of second order solution. The inner optimizer (LM) time is for the second loop, using the LM optimizer to find the minimum in second order.

When the size of the datasets are small, ($n \leq 500$), most of the time is spent on computing the second order solution. Only less than 10% of the time is cost by the first loop, computing the eigen pair. Most of the time is spent on the inner optimizer. While the number of camera poses grows, the percent of time taken by first order solution computation increased to about 30% but still less than the time spent on the inner optimizer. The increase percentage of eigen pair computation is related to the big matrix composed during the optimization process. To be more specific, the $\bar{L}$ and $S^{\intercal}$, in Equation 2.3. As the number of camera poses n increases, the computation performed on these $np \times np$ dimension matrix will grow in $O(n^2)$ both in space and runtime.

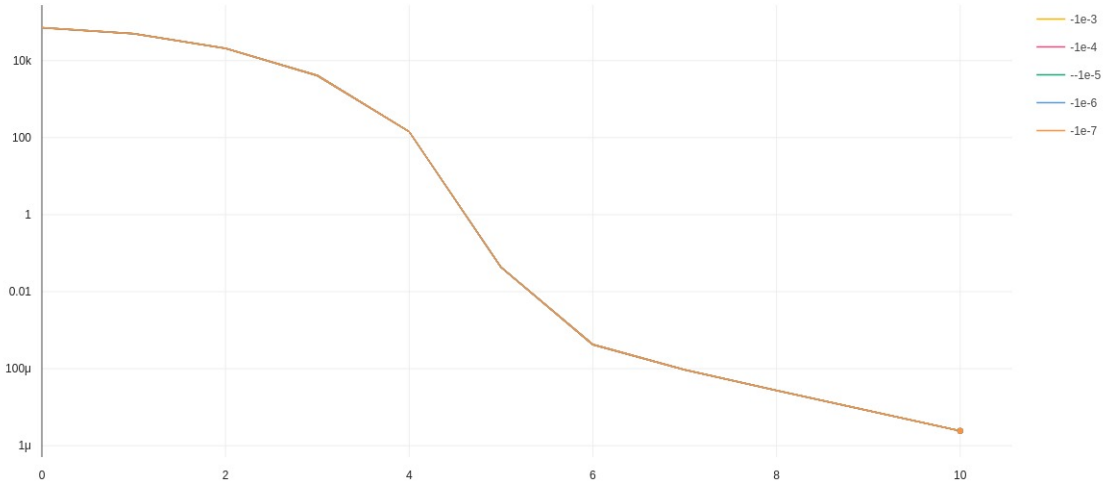
CHAPTER 4

THE INFLUENCE OF HYPER-PARAMETERS

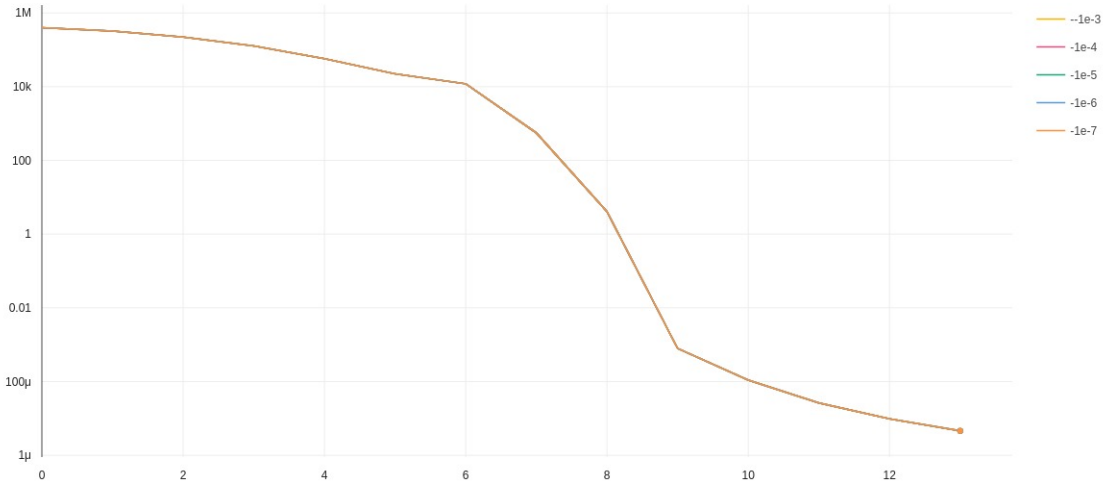
In this chapter, I study how the parameters in Shonan Averaging algorithm. There are two parameters, the optimal threshold for the eigen value (which is also the certification for checking global optimal) and the absolute and relative error threshold inside the LM optimizer.

4.1 Eigen value threshold

As observed in my experiment, Shonan Averaging algorithm usually find the global optimal solution at level 5 (lift up the rotation to $SO(5)$. Given the same random initialization, I tried different eigen value thresholds from $-1e^{-3}$ to $-1e^{-7}$, the ultimate Frobenius errors of the solutions returned are the same even after ten digits of the decimal point in Figure 4.1. The y-axis is the Frobenius error and the x-axis is the iteration times. This means during the optimization process, the eigen value certification from the first order solution will return a value less than $-1e-8$. Thus changing the eigen value threshold will not influence whether there will be another round of dimension lifting.

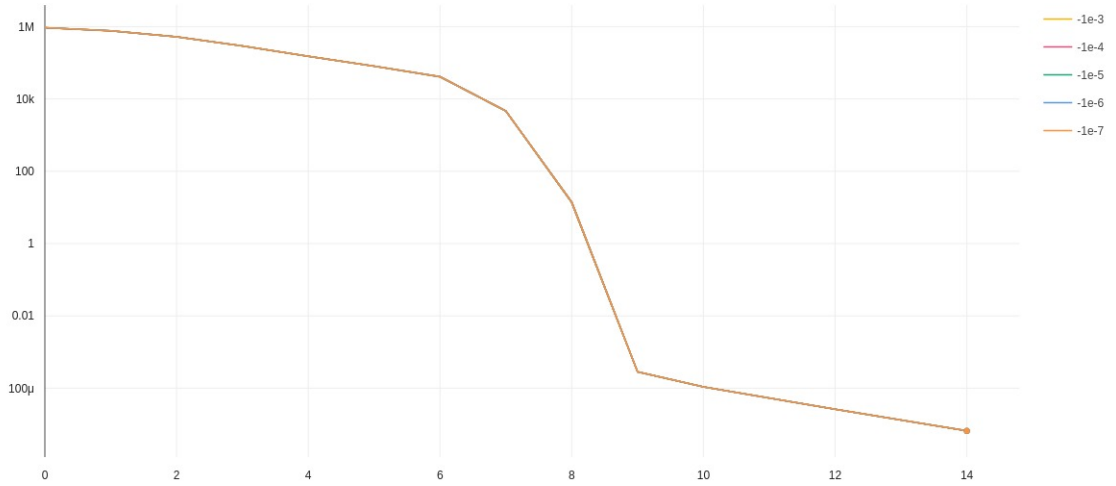


(a) natural_history_museum_london, n = 30, m = 274

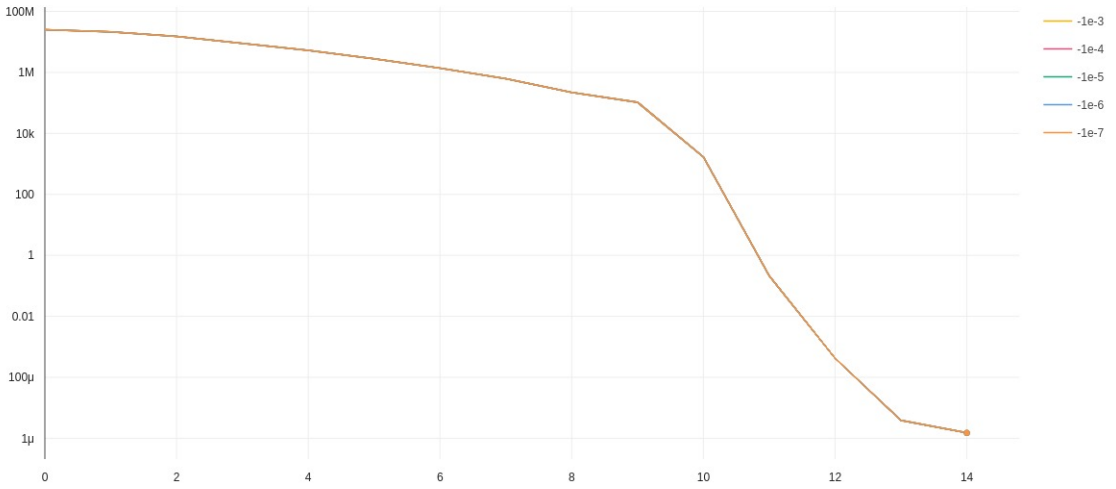


(b) london_bridge, n = 88, m = 1500

Figure 4.1: Performance of SA with eigen value threshold from -1e-3 to -1e-7 on YFCC datasets.



(c) lincoln_memorial, n =127, m = 3516



(d) notre_dame_rosary_window, n= 326, m = 93104

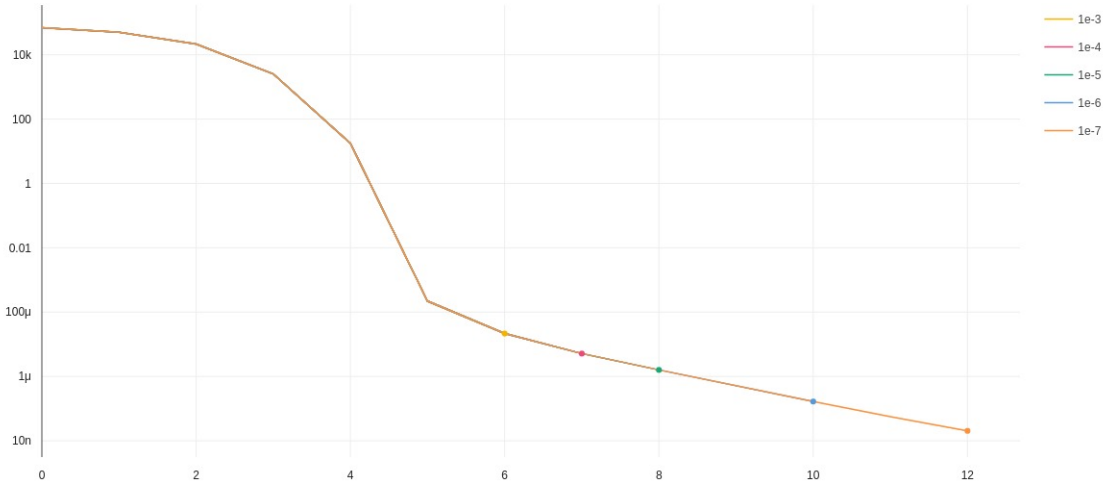
Figure 4.1: Performance of SA with eigen value threshold from -1e-3 to -1e-7 on YFCC datasets.

4.2 Absolute and relative error threshold inside LM optimizer

The parameters that will influence the accuracy of Shonan Averaging algorithm is the absolute error threshold and relative error threshold. The closer the thresholds are to zero the more accurate the solution will be. In the LM optimizer, these two thresholds are for com-

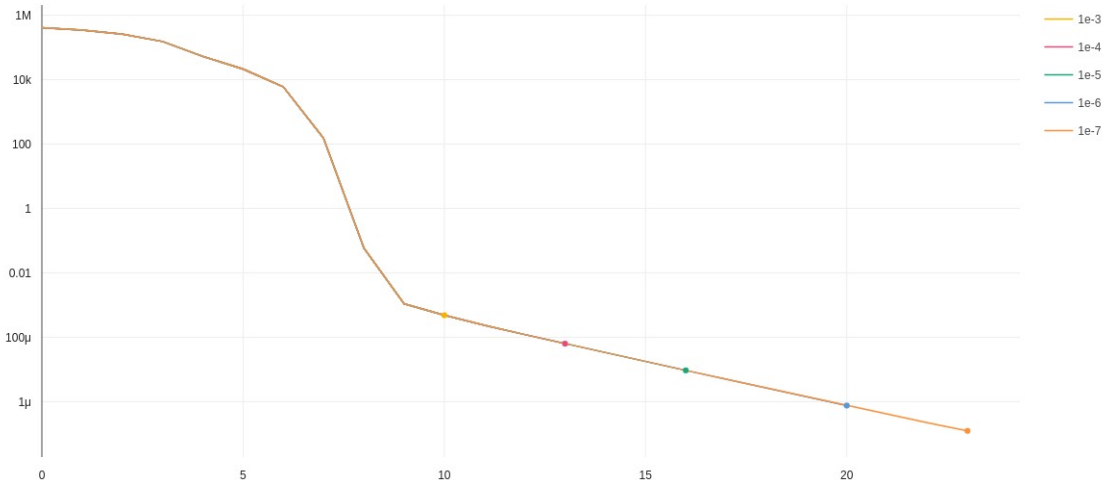
paring the solutions between two inner iterations inside LM optimizer. If the absolute error and relative error do not exceed the thresholds, then LM optimizer identify the solution as the minimum solution it can find. Thus the closer the thresholds are to zero, the more accurate solution it can find due to more iterations.

In the experiment shown in Figure 4.2, the y-axis is the Frobenius error and the x-axis is the iteration times, I tested different absolute and relative error threshold, the absolute error threshold is always equal to relative threshold for convenience. When the error thresholds become closer to zero, the deeper (closer to zero) the Frobenius error of the solution is. But the more iterations does not cost too much of extra time. Thus to generate a more accurate result can be achieved by tuning the error thresholds inside the LM optimizer.

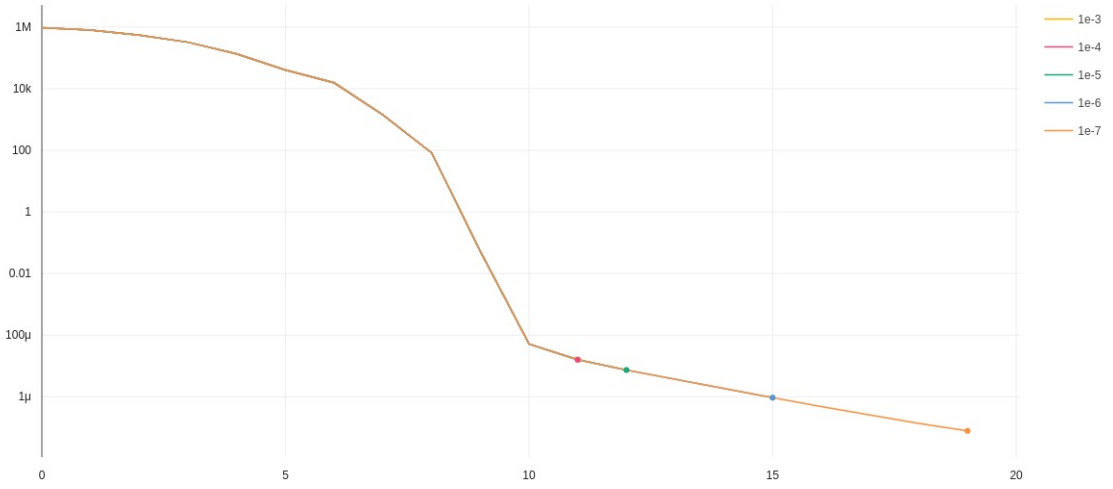


(a) natural_history_museum_london, n = 30, m = 274

Figure 4.2: Performance of SA with absolute error and relative error threshold from -1e-3 to -1e-7 on YFCC datasets.

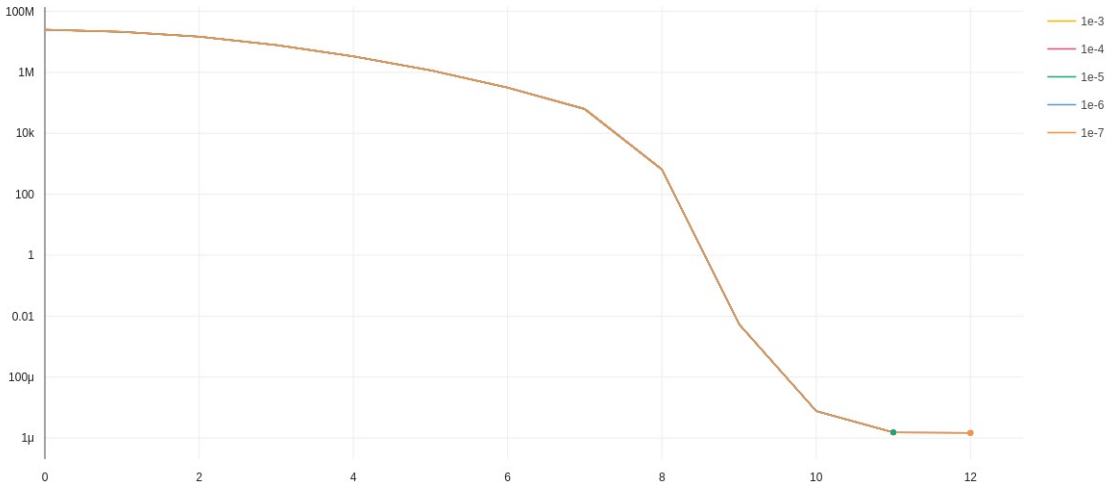


(b) london_bridge, n = 88, m = 1500



(c) lincoln_memorial, n =127, m = 3516

Figure 4.2: Performance of SA with absolute error and relative error threshold from -1e-3 to -1e-7 on YFCC datasets.



(d) notre_dame_rosary_window, n= 326, m = 93104

Figure 4.2: Performance of SA with absolute error and relative error threshold from -1e-3 to -1e-7 on YFCC datasets.

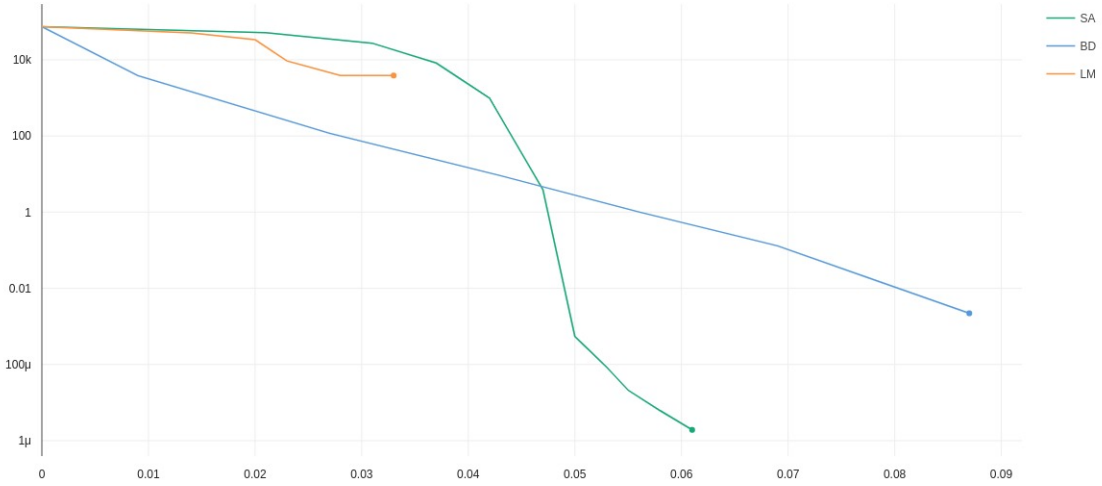
CHAPTER 5

COMPARISON WITH STATE OF THE ART

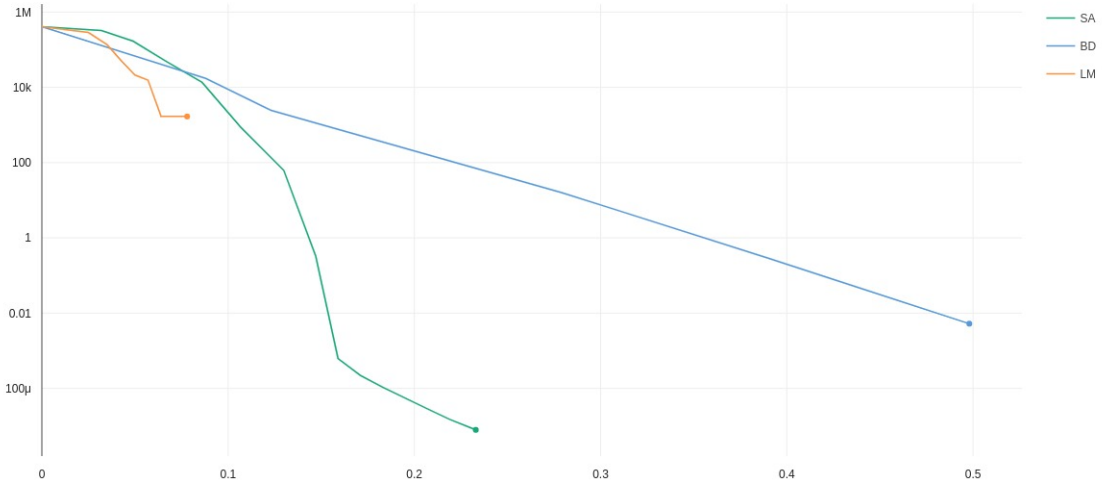
I evaluate the performance of Shonan Averaging by comparing with other optimization methods, Levenberg Marquardt and rotation averaging algorithm in Eriksson's paper [15] (block coordinate descent method) on several structure from motion datasets and Watts-Strogatz graph problems.

In the last section of the experiments, I compared the performance of Shonan Averaging (SA), Block coordinate Descent (BD) and Levenberg-Marquardt (LM) given the same initialization generated randomly. The results on different size of the datasets show Shonan Averaging always converges the fastest and is guaranteed to recover the solution with the minimum cost. Despite all parameters and possible improvements studied by previous part, Shonan Averaging algorithm is still the fastest algorithm comparing with the LM and BD method. Figure 5.1 and Figure 5.2 show the accuracy and speed performance of SA, BD and LM during optimization.

5.1 Random initialization on LM, BD and SA

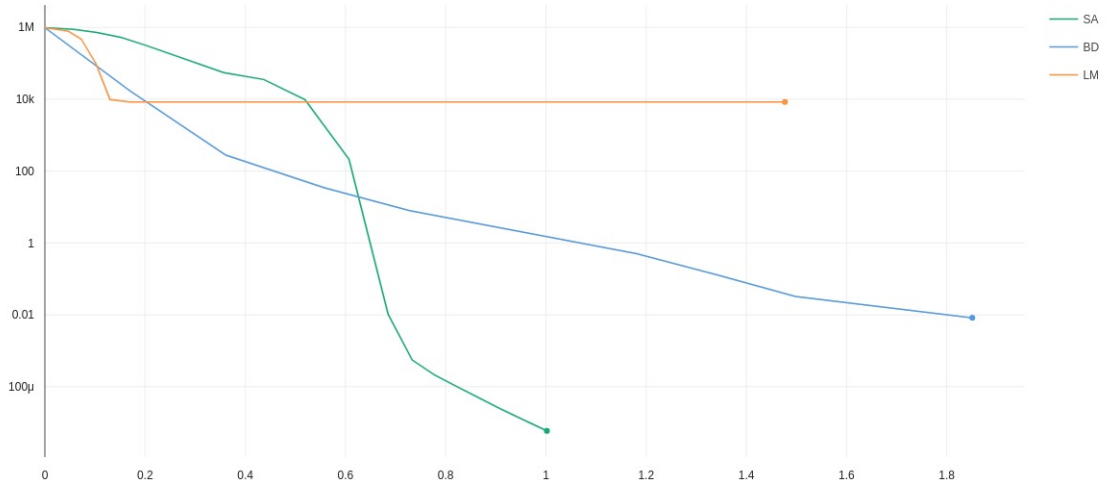


(a) natural_history_museum_london, n = 30, m = 274

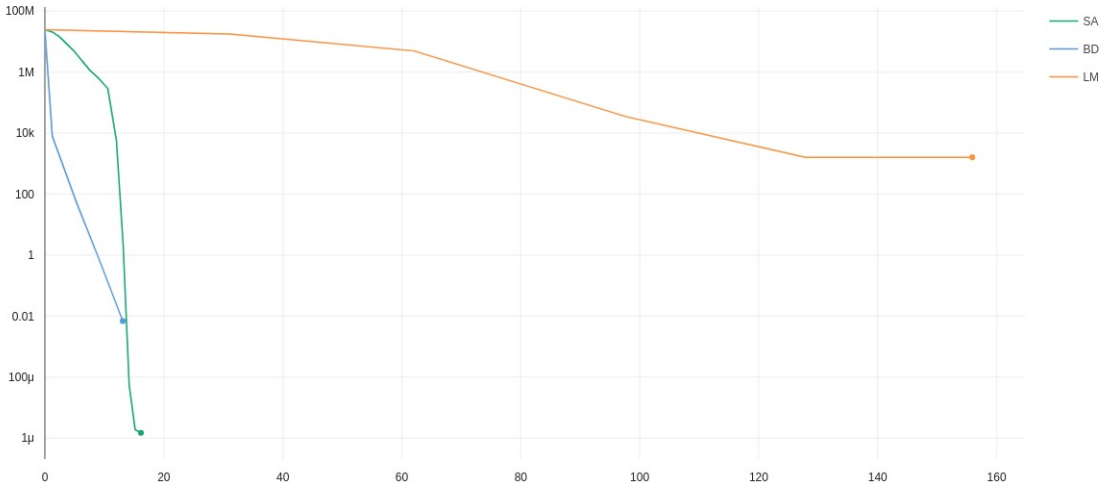


(b) london_bridge, n = 88, m = 1500

Figure 5.1: Performance of LM, BD, SA on YFCC datasets with random initialization.



(c) lincoln_memorial, n =127, m = 3516



(d) notre_dame_rosary_window, n= 326, m = 93104

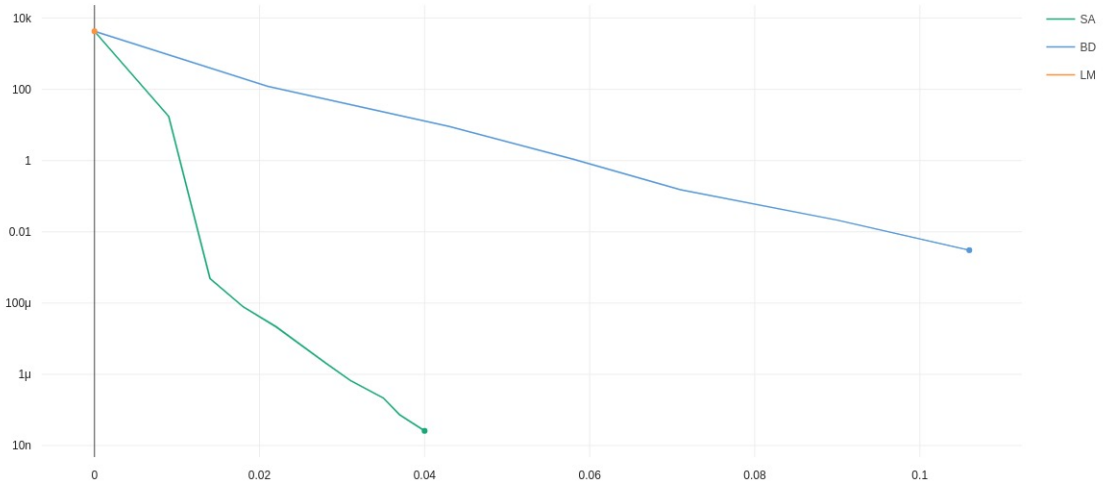
Figure 5.1: Performance of LM, BD, SA on YFCC datasets with random initialization.

Figure 5.1 shows the performance of SA, BD and LM given random initialization, the y-axis is the Frobenius error and the x-axis is the time in seconds. The size (number of camera poses) of the four datasets tested ranges from 30 to 326. With totally random initialized orientations, both methods start with a large Frobenius error, from $10K$ to $100M$. During optimization, the Frobenius error of LM method decrease extremely slow. For small dataset

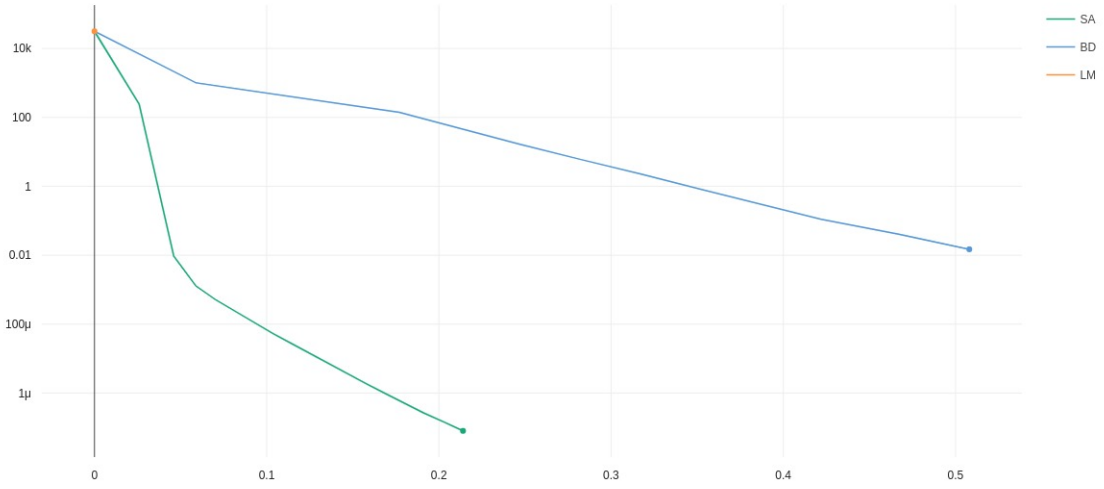
with camera pose number less than 100, it converges earlier than SA and BD but the error is unsurprisingly large; for large dataset with camera pose greater than 100, LM gives more iteration but still converges at a high error level except for one dataset. Since when the camera pose number increase, the accumulated error also increase, the accumulated error is not within the given error threshold (absolute error threshold) and the iteration inside LM continues until it satisfy the circumstance. Without the global optimal guarantee, LM stuck at a local minimum and continue to iterating from the local mimimum point. Though it costs more time or iteration, the solution found by LM is far away from the optimal. BD performs better than LM but not as good as SA. It gives better accuracy performance than LM but did not reach the level of SA possibly due to the rounding process in BD algorithm. The time spent by BD is much longer.

5.2 MST initialization on LM, BD and SA

Figure 5.2 shows the performance of SA, BD and LM given mst initialization, the y-axis is the Frobenius error and the x-axis is the time in seconds. SA and BD perform similar with mst initialization and random initialization. But we can see more clearly here that SA outperforms BD both in speed and accuracy on small datasets and large datasets. LM only has one iteration given mst initial estimation. For the mst initial estimation is already close to the ground truth, LM does not have the procedure for dealing with it because it only use the error as the forecast that the solution is a minimum point or not. Given the mst initialization, SA still beats BD and LM both in speed and accuracy.

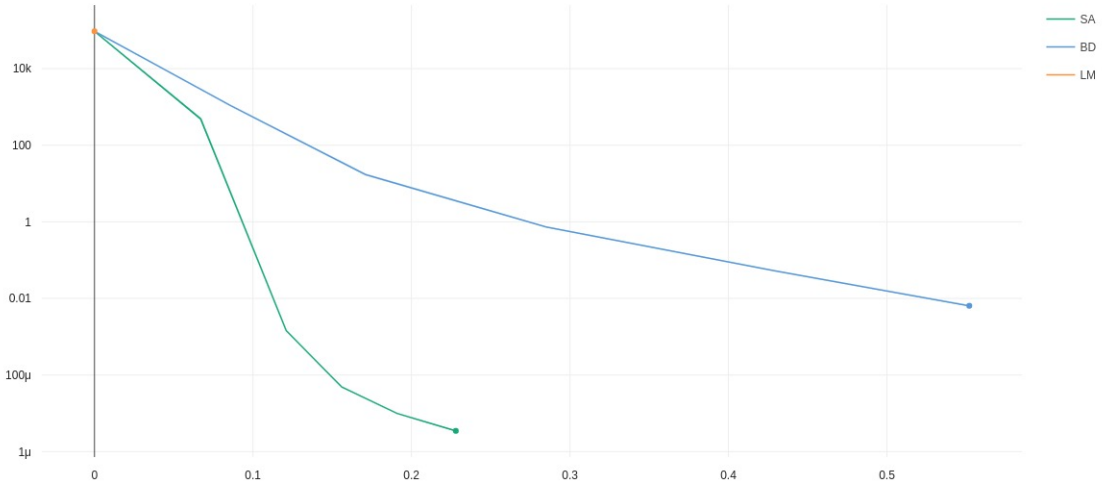


(a) natural_history_museum_london, n = 30, m = 274

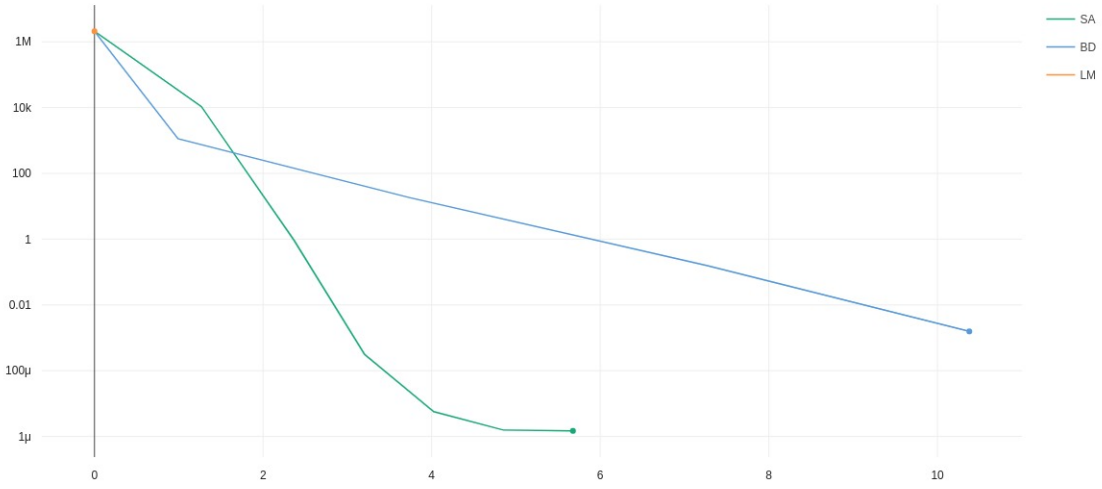


(b) london_bridge, n = 88, m = 1500

Figure 5.2: Performance of LM, BD, SA on YFCC datasets with mst initialization.



(c) lincoln_memorial, n =127, m = 3516



(d) notre_dame_rosary_window, n= 326, m = 93104

Figure 5.2: Performance of LM, BD, SA on YFCC datasets with mst initialization.

CHAPTER 6
DISCUSSION

6.1 Conclusion

In this dissertation, I have shown that Shonan Averaging algorithm provides a more accurate and faster way to give solution to rotation averaging algorithm while guarantee the solution to be global optimal instead of local optimal. This is of high importance in real world application like SfM or SLAM system especially for SfM system. In SfM system, the absolute camera poses are not provided. While in SLAM system, the absolute camera poses are computed incrementally and recover the absolute camera poses can be much easier. Thus an algorithm like Shonan Averaging can improve the accuracy of pose estimation and even the efficiency of the whole pipeline by providing a much more accurate and faster rotation estimation as long as the measurements are in low noise level.

6.2 Future Work

6.2.1 Inner optimization improvement

Though Shonan Averaging algorithm provides a fast and accurate way for solving rotation averaging algorithm. In the experiments, the timing performance of Shonan Averaging shows more space for improvement [32, 33]. Based on GTSAM, the process for computing the eigen pair can be factorized further and thus save the space and runtime for computing the eigen pair for certification. The process for computing the second order solution, which is the optimization process inside the LM optimizer, has more space to improve. Considering we only aim to solve the rotation averaging problem but we build a graph and use the optimizer inside the nonlinear factor graph to solve the second order solution, we can definitely improve the efficiency if only using nonlinear optimizer. In real world application,

the relative rotation information provided by visual odometry is usually not the only type of constraints, there are different kinds of sensors providing different kinds of constraints. But the graph we build can be used to handle other forms of constraint provided by sensors like Inertial Measurement Unit (IMU), the general performance can be more efficient if considering other type of measurements during optimization.

6.2.2 Accuracy performance improvement

We discuss in the chapter 3 that MST initialization can speed up the computation a lot comparing with random initialization. The datasets we test distribute the noise equally on the measurements while it is not the case in real world. In the future, constructing the weighted graph by taking the noise level into consideration is worth experimenting. With more structured initialization, Shonan Averaging might provide faster speed even though the speed and accuracy are already in an amazing level.

Appendices

APPENDIX A

EXPERIMENTAL EQUIPMENT

All experiments are tested by a computer with Intel Core i7-8550U @ 1.80GHz x 8 processor on Ubuntu 18.04 LTS. The timing result should be different with different properties.

APPENDIX B

DATA PROCESSING

Some of the graphs that show the Frobenius error change with time are generated by using https://www.comet.ml/, a Python package that provides a self-hosted and cloud-based meta machine learning platform allowing data scientists and teams to track, compare, explain and optimize experiments and models. This tool is extremely helpful when testing algorithm with many hyper parameters.

REFERENCES

[1] J. L. Schonberger and J.-M. Frahm, "Structure-from-motion revisited," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2016.

[2] V. M. Govindu, "Combining two-view constraints for motion estimation," 2001, pp. 218–225.

[3] K. Sim and R. Hartley, "Recovering camera motion using $l\infty$ minimization," vol. 1, 2006.

[4] S. Agarwal, N. Snavely, S. M. Seitz, and R. Szeliski, "Bundle adjustment in the large," in *European conference on computer vision*, Springer, 2010, pp. 29–42.

[5] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, "G2o: A general framework for graph optimization," in *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA)*, May 2011.

[6] S. Agarwal and K. Mierle, *Ceres solver: Tutorial & reference*, Google Inc., 2012.

[7] F. Dellaert, "Factor graphs and GTSAM: A hands-on introduction," Georgia Institute of Technology, Tech. Rep. GT-RIM-CP&R-2012-002, Sep. 2012.

[8] F. Dellaert, D. M. Rosen, J. Wu, R. Mahony, and L. Carlone, "Shonan rotation averaging: Global optimality by surfing $so(p)n$," in *European Conference on Computer Vision*, Springer, 2020, pp. 292–308.

[9] R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, Second. Cambridge University Press, ISBN: 0521540518, 2004.

[10] T. Cover and J. Thomas, *Elements of Information Theory*. 1991.

[11] L. Carlone, D. Rosen, G. Calafiore, J. Leonard, and F. Dellaert, "Lagrangian duality in 3D SLAM: Verification techniques and optimal solutions," 2015, pp. 125–132.

[12] D. Rosen, L. Carlone, A. Bandeira, and J. Leonard, "SE-Sync: A certifiably correct algorithm for synchronization over the Special Euclidean group," San Francisco, CA, Dec. 2016.

[13] K. Wilson and D. Bindel, "On the distribution of minima in intrinsic-metric rotation averaging," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 6031–6039.

[14] J. Fredriksson and C. Olsson, "Simultaneous multiple rotation averaging using lagrangian duality," 2012.

[15] A. Eriksson, C. Olsson, F. Kahl, and T.-J. Chin, "Rotation averaging and strong duality," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 127–135.

[16] K. Levenberg, "A method for the solution of certain non – linear problems in least squares," *Quarterly of Applied Mathematics*, vol. 2, pp. 164–168, 1944.

[17] D. J. Watts and S. H. Strogatz, "Collective dynamics of 'small-world'networks," *nature*, vol. 393, no. 6684, pp. 440–442, 1998.

[18] B. Thomee, D. A. Shamma, G. Friedland, B. Elizalde, K. Ni, D. Poland, D. Borth, and L.-J. Li, "Yfcc100m: The new data in multimedia research," *Communications of the ACM*, vol. 59, no. 2, pp. 64–73, 2016.

[19] B. Ayush, *Deep front ends*.

[20] A. Singer and H.-T. Wu, "Vector diffusion maps and the connection laplacian," *Communications on Pure and Applied Mathematics*, vol. 65, no. 8, pp. 1067–1144, 2012. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/cpa.21395.

[21] E. W. Swokowski, *Calculus With Analytic Geometry /*, 2d ed. Boston : Prindle, Weber Schmidt, c1979. Includes index.

[22] D. Rosen and L. Carlone, "Computational enhancements for certifiably correct slam," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2017.

[23] D. M. Rosen, L. Carlone, A. S. Bandeira, and J. J. Leonard, *A certifiably correct algorithm for synchronization over the special euclidean group*, 2017. arXiv: 1611.00128 [cs.RO].

[24] P.-A. Absil, R. Mahony, and R. Sepulchre, *Optimization Algorithms on Matrix Manifolds*. Princeton, NJ, USA: Princeton University Press, 2007.

[25] J. Nocedal and S. J. Wright, *Numerical Optimization*, ser. Springer Series in Operations Research. Springer-Verlag, 1999.

[26] R. E. Greene and S. G. Krantz, *Function theory of one complex variable*. American Mathematical Soc., 2006, vol. 40, p. 189.

[27] R. A. Reyment and K. Jvreskog, *Applied factor analysis in the natural sciences*. Cambridge University Press, 1996, p. 68.

[28] D. Koller and N. Friedman, *Probabilistic graphical models: principles and techniques*. MIT press, 2009.

[29] K. Wilson, D. Bindel, and N. Snavely, "When is rotations averaging hard?" In *European Conference on Computer Vision*, Springer, 2016, pp. 255–270.

[30] R. Diestel, A. Schrijver, and P. Seymour, "Graph theory," *Oberwolfach Reports*, vol. 7, no. 1, pp. 521–580, 2010.

[31] L. Carlone, R. Tron, K. Daniilidis, and F. Dellaert, "Initialization techniques for 3d slam: A survey on rotation estimation and its use in pose graph optimization," in *2015 IEEE international conference on robotics and automation (ICRA)*, IEEE, 2015, pp. 4597–4604.

[32] Á. Parra, S.-F. Chng, T.-J. Chin, A. Eriksson, and I. Reid, *Rotation coordinate descent for fast globally optimal rotation averaging*, 2021. arXiv: 2103.08292 [cs.CV].

[33] Y. Chen, J. Zhao, and L. Kneip, *Hybrid rotation averaging: A fast and robust rotation averaging approach*, 2021. arXiv: 2101.09116 [cs.CV].